Recognition of R-Metrics

Thomas Gatter David Schaller Carsten Seemann Peter F. Stadler

January 11, 2022

1 Introduction

The genomes of higher eukaryotes typically contain many families of genes with similar DNA sequence. These usually encode similar proteins and share similar function. Their sequence similarity indicates that they have evolved from a single original ancestor by means of multiple rounds of duplication. Such paralogous genes are often, but by no means always, located at the same genomic locus, where they form a gene cluster. In many cases clustered genes are not tied together functionally and the clusters can disintegrate by genome rearrangement without detrimental effects.

The details of the molecular mechanisms and evolutionary forces that govern the expansion of clusters of paralogous genes are by no means completely understood.

The classic model assumes that single genes are duplicated as complete and faithful copies. Each gene cluster is generated by repeated duplication steps. Differences in sequence and function arise as each locus evolves independently over time. Unequal crossing over, as a possible molecular mechanism for duplication, occurs as a special case of crossing over. Here, genetic material of two sister chromatids or homologous chromosomes is exchanged, but at misaligned genomic positions (Fig. 1(a)).

Walter J. Gehring, a developmental biologist famous for his studies of the Hox gene cluster in *Drosophila melanogaster*, interpreted the fact that the three Hox genes (abd-B, abd-A, and Ubx) appear in a tandem arrangement as evidence for unequal crossing over within the boundaries of genes. In this scenario, a gene copy is created as a hybrid of its left and right neighbors (Fig. 1(b)). We call distance matrices of gene clusters formed by this mechanism *Type R distance matrices* or short *R-maps*.

In this practical course, we want to further characterize the properties of *R-maps* and explore algorithms towards their recognition.

2 Definitions and Simulation

Given a real life biological sample, we can only estimate distances based on pairwise sequence similarity, e.g. based on an alignment. Therefore we gain a 2-dimensional matrix of distances.

More formally, we represent distances between genes as a *distance map d on X* as $d : X \times X \to \mathbb{R}_{\geq 0}$, which is

1. symmetric, i.e., $d(x, y) = d(y, x)$ for every $x, y \in X$, and
2. positive semi-definite, i.e., $d(x, y) \geq 0$ and $d(x, x) = 0$ for every $x, y \in X$.

We call elements in X *leaves*. For three (not necessarily pairwise distinct) leaves $x, y, z \in X$ we set the *spike length* as follows

$$\Delta_d(x, y, z) \coloneqq \frac{1}{2}\Big(d(x, z) + d(y, z) - d(x, y)\Big).$$

Moreover, d is a *pseudo-metric* if d satisfies the "triangle inequality", i.e. $d(x, y) \leq d(x, z) + d(y, z)$ for every $x, y, z \in X$. Hence, $0 \leq d(x, z) + d(y, z) - d(x, y)$ for all $x, y, z \in X$. Shortened and without proof for this project description, we make the following

Observation 2.1. A distance map d on X is a pseudo-metric if and only if $\Delta_d(x, y, z) \geq 0$ for every $x, y, z \in X$.

Lemma 2.2. *A pseudo-metric d on X is additive if and only if every pairwise distinct $x, y, z, u \in X$ satisfy the "4-point condition", i.e., the four leaves can be renamed such that*

$$d(x, y) + d(z, u) \leq d(x, u) + d(y, z) = d(x, z) + d(u, y). \tag{1}$$

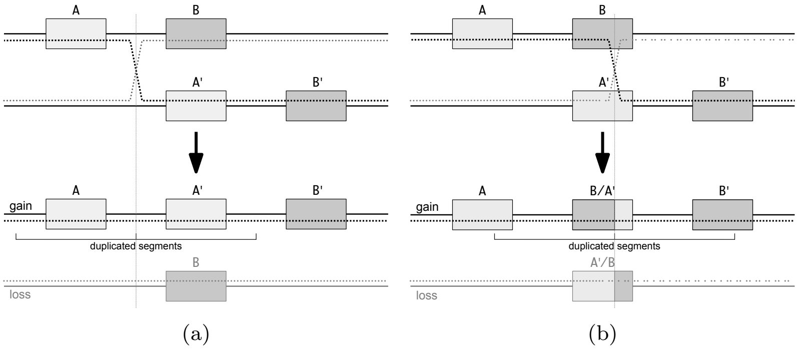
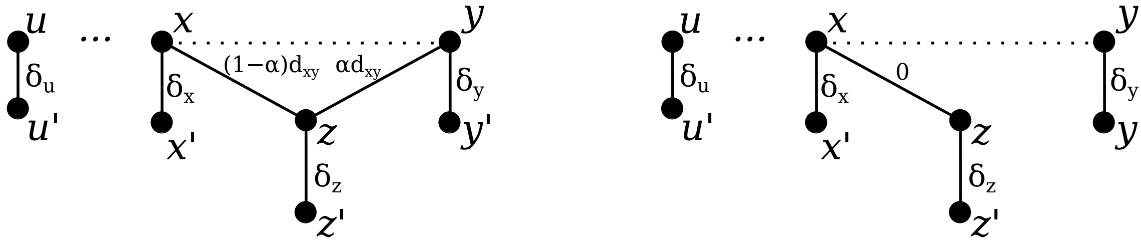


Figure 1: Gene cluster expansion by local gene duplication (a) and unequal crossing over in Gehring's model (b). During mitosis or meiosis, when chromatids are paired, unequal crossing over leads to a tandem duplication on one chromatid and a deletion on the sister chromatid or homologous chromosome, respectively. The loss of whole genes is considered to be lethal. In Gehring's model the crossing over occurs within the gene sequences resulting in hybrid genes. Crossing over between intergenic sequences results in duplication of complete genes.



(a) *merging* R-step

(b) *branching* R-step

Figure 2: Illustration of distances in an $R - Step$.

R-maps allow events in which the leaves themselves are recombined, i.e., instead of assuming that the newly introduced leaf z is a true copy of x (as in Fig. 1(a)), we now allow that z is a recombinant of two leaves x and y (Fig. 1(b)). To this end, we need the following:

Definition 2.3 (R-step)**.** Let X be a set of leaves, and let d be a distance map on X. Then, a map d' can be obtained by an *R-step* as defined as follows:

(R1) Create a new leaf $z \notin X$.

(R2) Define intermediate distances ξ as follows

a) choose two (not necessarily distinct) leaves $x, y \in X$, called *parents (of z)*,
b) choose an R-step specific ratio $\alpha \in [0,1]$, and
c) set

$$\begin{aligned} \xi_{zu} &:= \alpha d_{xu} + (1-\alpha) d_{yu} \quad \text{for all } u \neq x, y, z \\ \xi_{zx} &:= (1-\alpha) d_{xy} \\ \xi_{zy} &:= \alpha d_{xy} \end{aligned}$$

(R3) Define an updated map $d' : X' \times X' \to \mathbb{R}$ with the new leaf $X' = X \cup \{z\}$ by

a) choosing for each leaf $p \in X'$ an individual evolutionary rate $\delta(p) \in \mathbb{R}_{\geq 0}$, and
b) setting, for all $p, q \in X'$,

$$d'_{pq} := \begin{cases} 0 & \text{if } p = q, \\ \xi_{pq} + \delta(p) + \delta(q) & \text{else if } z \in \{p, q\}, \\ d(p,q) + \delta(p) + \delta(q) & \text{else.} \end{cases} \tag{2}$$

In this case, we say that the map d' is obtained by an R-step $(x, y : z)_\alpha$ applied on d. An R-step $(x, y : z)_\alpha$ is a *merging* R-step if $x \neq y$ and $\alpha \in]0,1[$, and a *branching* R-step, otherwise.

Now, we define R-maps.

Definition 2.4 (R-maps)**.** A distance map d' on X' is of *type R* or shortly, an *R-map*, if

1. d' is a distance map on one leaf, i.e. $|X'| = 1$, or
2. d' is obtained by an R-step applied on an R-map d on $X := X' \setminus \{z\}$ for some leaf $z \in X$.

This definition also yields a construction algorithm to simulate R-maps. We may further classify scenarios by three properties:

(P1) The definition given above allows the creation of branching steps where a gene is exactly duplicated. For this project we restrict analysis to scenarios where $\alpha \in]0,1[$, and therefore, with the exception of the first R-step (that is applied on a single leaf), only merging R-steps are applied thus generating only hybrid genes.

(P2) The definition above puts no restriction on the choice of parents $x, y \in X$. The unequal crossing over model used as motivation for this metric requires that parents are strict genomic neighbors, thus bar the introduction of further genome rearrangement steps, diverges from this definition. We therefore define a restricted model called *circular*, were we regard the gene cluster to be organized on a circular chromosome. Thereby we define a circular order of all genes. x, y have to be chosen such that they are neighbors, and the new leaf z is inserted in between.

(P3) Step (R3) assumes that each leaf/gene evolves at unique (non-linear) pace, i.e. mutation rates and their ratios between genes varies for each step. From a biological standpoint, closely related genes often evolve at a similar rate. We therefore may choose a common evolutionary rate δ that is applied to all genes. In this scenario δ constitutes the time between duplication events in the sense of a biological clock. We therefore call this scenario *clocklike.*

We can characterize R-maps as follows:

Lemma 2.5. *A distance map d is additive if and only if d is an R-map, where each R-step was a branching step.*

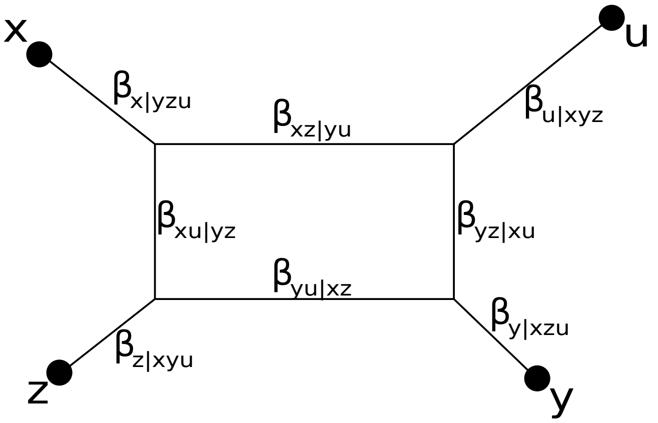


Figure 3: The figure shows the box-graph on the leaf-set $X = \{x, y, z, u\}$, which is a graph G, where for each edge $e \in E$ a non-negative weight $\beta_e \in \mathbb{R}_{\geq 0}$ is assigned such that the opposite edges of the rectangle have the same lengths. Note that the sum of weights along a shortest path between to leaves is always the same.

Corollary 2.6. *Every additive pseudo-metric is an R-map.*

Lemma 2.7. *Every R-map is a pseudo-metric.*

We consider R-maps with three leaves as the smallest interesting case, constituting exactly two parents created by an initial branching R-step and a child created by single subsequent merging R-step. We can characterize R-maps with three leaves as follows:

Lemma 2.8. *Let d be a distance map on $X = \{x, y, u\}$ with three leaves. Then, the following statements are equivalent:*

1. *d is an R-map;*
2. *d is a pseudo-metric;*
3. *d is additive; and*
4. *there are $a, b, c \in \mathbb{R}_{\geq 0}$ such that $d(x, y) = a + b$, $d(x, u) = a + c$ and $d(y, u) = b + c$.*

Increasing the size of maps to four leaves, we consider a map d on $X = \{u, x, y, z\}$. Four leaves define six pairwise distances $d(x, y)$, $d(x, z)$, $d(x, u)$, $d(y, z)$, $d(y, u)$, $d(z, u)$, and the following three *distance sums* on four leaves:

$$d(x, y) + d(u, z)\,, \quad d(x, u) + d(y, z) \quad \text{and} \quad d(x, z) + d(u, y). \tag{3}$$

Moreover, d on 4 leaves can be *presented* by the so-called "box-graph" , which is illustrated in Fig. 3. The relation between metrics on four leaves and box-graphs is captured by the following results:

Lemma 2.9. *Let d be a metric on $X = \{u, x, y, z\}$. Then, the following two statements are equivalent:*

1. *$d(x, y) + d(u, z)$ is one of the largest distance sums of Equ. (3); and*
2. *the pairs $\{x, y\}$ and $\{z, u\}$ of leaves are antipodal (diagonal to each other) in the box-graph which represents d.*

Moreover, if d satisfies one of the previous statements, then the lengths of the box-graph are

$$\beta_{x|yzu} = \frac{1}{2}\Big(d(x, z) + d(x, u) - d(u, z)\Big) = \Delta_d(u, z, x);$$

$$\beta_{y|xzu} = \frac{1}{2}\Big(d(y, z) + d(y, u) - d(u, z)\Big) = \Delta_d(u, z, y);$$

$$\beta_{z|xyu} = \frac{1}{2}\Big(d(x, z) + d(y, z) - d(x, y)\Big) = \Delta_d(x, y, z);$$

$$\beta_{u|xyz} = \frac{1}{2}\Big(d(x, u) + d(y, u) - d(x, y)\Big) = \Delta_d(x, y, u);$$

$$\beta_{xz|yu} = \frac{1}{2}\Big(\big(d(x, y) + d(u, z)\big) - \big(d(x, z) + d(y, u)\big)\Big); \text{ and}$$

$$\beta_{xu|yz} = \frac{1}{2}\Big(\big(d(x, y) + d(u, z)\big) - \big(d(x, u) + d(y, z)\big)\Big).$$

Theorem 2.10. *Let d be a distance map on $X = \{u, x, y, z\}$. Then, the following three statements are equivalent:*

1. *d is an R-map.*
2. *d is a pseudo-metric, and if the largest distance sum of Equ.* (3) *is unique, say $d(x, y) + d(z, u)$, then we have*

   a) $\beta_{xu|yz} \cdot \beta_{xz|yu} \leq \beta_{x|yzu} \cdot \beta_{y|xzu}$ *or*
   b) $\beta_{xu|yz} \cdot \beta_{xz|yu} \leq \beta_{u|xyz} \cdot \beta_{z|xzy}$*,*

   where the βs are the lengths of the box-graph w.r.t. d.
3. *d is a pseudo-metric, and if the largest distance sum of Equ.* (3) *is unique, say $d(x, y) + d(z, u)$, then we have*

   a) $d(x, y) \cdot \Big(d(x, y) + 2 \cdot d(z, u) - \zeta\Big) \leq \Big(d(x, z) - d(y, z)\Big) \cdot \Big(d(y, u) - d(x, u)\Big)$*, or*
   b) $d(z, u) \cdot \Big(d(z, u) + 2 \cdot d(x, y) - \zeta\Big) \leq \Big(d(x, z) - d(x, u)\Big) \cdot \Big(d(y, u) - d(y, z)\Big)$*,*

   where $\zeta := d(x, z) + d(y, u) + d'(x, u) + d(y, z)$.

Moreover, Statement (2a) and (3a) are equivalent, and imply that x and y are the parents of the fourth leaf. Analogously, Statement (2b) and (3b) are equivalent, and imply that z and u are the parents of the fourth leaf.

Loosely speaking, Condition 2 in Thm. 2.10 says that, for an R-map, at least one pair of spikes opposite to each other in the box-graph must not be too short. In particular, they must be longer the bigger the area of the rectangle is.

3 Reconstruction of R Maps

Given a distance matrix d from either real life data or simulation, we are interested in particular in its reconstruction, i.e:

1. the decision if d is an R-map
2. a sequence of R-steps explaining the construction of the matrix

In the original paper by Prohaska et al. 2017, Alg. 1 was proposed to solve both parts. However, this algorithm falls short on several aspects:

i) it is implied that every metric on four leaves is an R-map, which is not true.

ii) it does not check whether the resulting map is still a pseudo-metric.

iii) it does not check whether all δ are non-negative.

iv) it does not explicit state, what happens with $\delta(x)$ or with $\delta(y)$ if $\alpha = 0$ or $\alpha = 1$, respectively.

For the sake of this project, we will disregard iv) and assume our model only consists of merging R-steps. Thm. 2.10 part 3 yields a simple test on 4 leaves to verify they conform to an R-map. We may apply this test on the remaining nodes after execution of Alg. 1 in accordance with i). A pseudo-metric test ii) is equally trivial to implement, verifying that the triangle inequality holds for every possible (sub-)sets of 3 leaves. As δ is explicitly reconstructed by Alg. 1 for each leaf, a test for iii) is equally trivial. However, even the inclusion of such additional tests does not guarantee correct results.

Each iteration of the algorithm may yield multiple triple candidates defining a valid α from which one is chosen arbitrarily. The remaining candidates do not in general remain valid options for a consecutive step as distances are modified. We experimentally observed that candidates may be chosen such that Alg. 1 ends in a dead-end, i.e., a branch without a valid solution, despite the existence of a valid decomposition into R-steps. As an exponential number of paths exists in a decision tree, enumerating all viable combinations of R-steps is prohibitively slow both in theory and practice for maps with large dimensions. Branch-and-bound like strategies can be devised to minimize the search space. Nevertheless we aim to develop a more effective solution.

Algorithm 1 Recognition of R-maps

```
Require: Distance matrix $\mathbf{D}'$, $n = |V| \geq 4$
  repeat
    for $(x, y, z) \subseteq V$ do
      for $\{u, v\} \subseteq V \setminus \{x, y, z\}$ do
        $\alpha = \frac{(d'_{uz}+d'_{vy})-(d'_{vz}+d'_{uy})}{(d'_{ux}+d'_{vy})-(d'_{vx}+d'_{uy})}$
      end for
      if $\alpha \in [0, 1]$ is the same for all $u, v$ then
        if $\alpha \neq 0, 1$ then
          $\delta_z = \frac{d'_{xz}+d'_{yz}-d'_{xy}}{2}$
          $d_{xy} = \frac{(d'_{uz}-\alpha d'_{ux}-(1-\alpha)d'_{uy})-2\delta_z+\alpha d'_{xz}+(1-\alpha)d'_{yz}}{2\alpha(1-\alpha)}$
          $\delta_x = d'_{xz} - (1 - \alpha)d_{xy} - \delta_z$
          $\delta_y = d'_{yz} - \alpha d_{xy} - \delta_z$
          $\delta_u \leftarrow 0$ for $u \in V \setminus \{x, y, z\}$
          compute $\mathbf{D}$ as $d_{pq} = d'_{pq} - \delta_p - \delta_q$ for all $p, q \in V$
        end if
        $\mathbf{D}' \leftarrow \mathbf{D}$ without row and column $z$
        $n \leftarrow n - 1$
      end if
    end for
    if no $(x, y, z)$ was found then
      return false
    end if
  until $n = 4$
  return true
```

4 Workpackages

4.1 WP0: Preparation

- Familiarize yourself with the definitions presented in this project description and the secondary literature.
- Download and install the `Erdbeermet` tool from `https://github.com/david-schaller/Erdbeermet`. Ensure that all examples given in the tool description run without errors and make sure you understand all parts of the README.
- As a part of this project the file *src/erdbeermet/recognition.py* has to be modified or re-implemented (see below). Accordingly, familiarize yourself with its contents and how they relate to the theory presented in this project description and the secondary literature.

4.2 WP1: Simulation

As a first step, we need to develop a general classification pipeline that generates a distance matrix and feeds it into the recognition algorithm. The following requirements should be met:

- Simulation should cover a variety of matrix sizes and include default, as well as (combinations of) circular and clockwise scenarios (see parameters of the function `simulate()`). **Ensure that branching probability is 0.**
- **For every combination of simulation parameters, datasets at a magnitude of greater than 20,000 must be created and tested for each WP.**
- Prepare that your pipeline allows to pass not only the simulated distance matrix but also additional parameters such as the first 4 leaves in the generation process to the recognition algorithm (see below).

- Include a simple mechanism to change parameters and methods of the recognition algorithm. Modifications implemented in the following workpackages should be available as parameter switches or separate methods without e.g. commenting in/out code parts in the recognition algorithm.

Include an evaluation mechanism capable of the following:

- Classify whether the distance matrix was correctly recognized as an R-Map.
- Classify whether the final 4-leaf map after recognition matches the first 4 leaves of the simulation.
- Measure the divergence of the reconstructed steps from true steps of the simulation, e.g. by counting common triples **regardless of order**. **You may additionally analyze the divergence in order of common R-Steps**.
- Measure average runtimes.
- Output or respectively plot distance matrices, recognition steps and final box plots of scenarios, e.g. when reconstruction fails.

4.3 WP2: Exploration of original algorithm as a base metric

Use the classification pipeline of WP1 to explore the extended base-algorithm as implemented by `Erdbeermet` using *first_candidate_only=True*. **Out of the list of candidates for the last step choose a random, positive one if it exists. Use this option also for WP3 and WP4.** WP3 and WP4 later should be run on the same set of simulated matrices to highlight improvements in the chosen benchmarks, e.g. the frequency of failed recognition, the detection of co-optimal solutions and divergence from simulated R-steps.

4.4 WP3: Recognition with blocked leaves

We hypothesize that the Alg. 1 cannot run into a dead-end if candidates are chosen such that non of the first 4 or 3 leaves of the simulation are reduced as z. To test this conjecture modify the algorithm as follows:

- Modify the algorithm such that a set of leaf-identifiers B can be passed as a parameter.
- During execution no $z \in B$ may be chosen.
- Write a wrapper function that passes the first 4 or respectively 3 leaves of the simulation as B and benchmarks results as in WP2.
- Assume the realistic case where the core leaves are not known. Write a wrapper that iterates through all subsets of 4 or respectively 3 leaves until an R-map was correctly identified. Benchmark as in WP2.

4.5 WP4: Recognition by smallest spike consumption

We note that Alg. 1 cannot run into a dead end if we follow the exact evolutionary path, i.e. the sequence of R-steps to generate the map, in inverse order. As a second hypothesis, we presume that we can identify the operation that occurred last during generation, and therefore has to be reconstructed first, can be identified as it exhibits the "smallest" spike-length of all valid candidates. To test this conjecture modify the algorithm as follows:

- Identify all candidate triples $(x, y, z) \subseteq V$ with valid α as before.
- Then compute the next candidate
- Compute a minimal candidate respective to spike lengths as follows:
  - For each triple (x, y, z) with corresponding α compute the spike lengths δ_x, δ_y, and δ_z.

To this end, let $\Delta(a,b,c) \coloneqq \frac{1}{2}\big(d(a,c)+d(b,c)-d(a,b)\big)$.
The spike length that we need can be computed as

$$\begin{aligned}
\delta_z &= \Delta(x,y,z) \\
\delta_x &= \Delta(u,y,x) - \frac{1}{2\alpha}\big(d(x,y)+d(z,u)-d(x,z)-d(y,u)\big) \\
\delta_y &= \Delta(u,x,y) - \frac{1}{2(1-\alpha)}\big(d(x,y)+d(z,u)-d(x,u)-d(y,z)\big)
\end{aligned}$$

for some arbitrary $u \in X \setminus \{x,y,z\}$, i.e., an arbitrary leaf among the remaining ones that is furthermore distinct from x, y, and z.

- **Alternatively:** These spike length may be computed by `Erdbeermet` using the function `_compute_deltas(...)` which is already called in l. 231 and l. 400 of file 'recognition.py' (`Erdbeermet` version 0.0.4). You can directly use the returned values `delta_z, d_xy, delta_x, delta_y`, where `d_xy` is not needed.
- **Note**: `Erdbeermet` computes δ_x, δ_y, and δ_z using an alternative approach described by Prohaska et al. 2017 (`https://link.springer.com/article/10.1007/s00285-017-1197-3`).
- We define $U = (u_1, u_2, u_3) \prec U' = (u'_1, u'_2, u'_3)$, if $u_i = u'_j$ and $\delta_{u_i} < \delta_{u_i}$
(Candidate U evolved later then U' if it has a shorter spike for at least one shared leaf)
- Find a candidate for which no other candidate is smaller. If multiple such candidates exist, chose an arbitrary one of those.
  - You may implement this as a graph as described in the video. However, as we are only interest in all candidates that have no smaller candidates by pairwise comparison, all-vs-all comparison is sufficient, e.g. using a map to count results.
  - ~~We presume that for R-maps no circular relationships, e.g. $U \prec V, V \prec U$, can occur and a minimal candidate always exists.~~
  - **You may encounter cycles. If a minimal candidate nevertheless exists choose this candidate and resume as normal.**
  - **If no minimal candidate exists, i.e. all candidates are part of one or multiple cycles, stop the recognition and report a fail.**
  - **If no decomposition to R-steps can be found after reduction to 4 leaves, also report a fail.**
  - **Spike analysis only needs to be performed if more at least 6 leaves remain in the map ($n > 5$ in the code).**

- Continue the algorithm as before.
- Benchmark as in WP2.

Please refer also to the video enclosed to this task.

4.6 Summary and Presentation

Summarize all benchmarks for your final presentation.